JOURNAL OF COMPUTER AND SYSTEM SCIENCES 22, 198–219 (1981)

Interdependent Translation Schemes

M. P. GEORGEFF

School of Mathematical Sciences, The Flinders University of South Australia, Bedford Park, South Australia, 5042, Australia

Received October 11, 1978; revised November 6, 1980

A translation scheme for specifying translations which depend on both syntactic (input) structure and semantic (output) structure is introduced. Such schemes are called interdependent translation schemes. The syntactic and semantic structures are assumed to be context-free, but their interaction gives rise to a contextual dependence which provides considerably more power than the usual syntax directed translation scheme. It is shown that under two forms of leftmost restriction the class of translations generated by interdependent translation schemes is equal to the class of Turing-computable relations. Length increasing schemes generate translations which are characterized by context-sensitive languages. Interdependent translation schemes which allow associations between nodes in the input and output derivation trees are introduced, and it is shown that these also generate all Turing-computable relations. Normal forms for these schemes are also defined.

1. INTRODUCTION

The formal specification of translations between languages is of considerable importance in areas such as compiler design and natural language processing. Most of the devices so far developed [2–4, 6, 8–11, 17] have been syntax directed, that is, a translation is determined by the structure of the input language. Furthermore, this structure is assumed to be context-free.

Intuitively, one obtains a particular syntax directed translation by reordering elements appearing in the derivational structure of the input word and by relabelling, inserting or deleting terminal elements, all according to fixed rules. The scheme can be generalized to allow multiple output elements to be associated with a single input element [1]. Alternatively, syntax directed translation schemes can be viewed as gsm maps on trees [12, 14, 15].

However, all these schemes suffer from two major limitations. Firstly, all are syntactic mappings in the sense that the translation depends only on the structure of the input language. An immediate consequence of this is that the grammar describing the output language must be almost identical to that of the input language. However, in most practical applications, especially those involving natural languages, the structure of the input and output languages is quite dissimilar.

The second major limitation is that syntax directed translation schemes are not very powerful. In all cases the domain of the translation must be a context-free

0022-0000/81/020198–22$02.00/0

language, and this condition is violated for the majority of natural languages and even for most programming languages (e.g., [7]). Further, for a given translation, the derivational structure of each output word can only be a very simple permutation of the derivational structure of the corresponding input word.

To overcome these limitations, one could generalize the notion of a syntax directed translation scheme by allowing the structure of the input language to be non context-free. However, even under this generalization, the class of translations that can be specified is limited. Other generalizations exist (e.g., [16]), but again, the power of these translation schemes is limited and, from a linguistic point of view, they are of little interest.

In this article we consider an alternative approach in which the translation depends on the structure of both the input and output languages. Intuitively, the occurrence of a particular element in the derivational structure of an input word specifies the occurrence of a corresponding element in the derivational structure of the output word. However, these elements need not be similar, nor need the derivational structures be similar.

We will restrict out attention to the case where the derivational structures of the input and output languages are, taken separately, context-free. However, in the generation of a particular translation, the derivational structure of the input word will depend on the appearance of certain elements in the derivational structure of the output word, and vice versa. Thus the interaction between the two derivational structures gives rise to contextual dependence, and neither the domain nor the range of the translation is restricted to being a context-free language.

We will call such schemes interdependent translation schemes (ITSs). In Section 2 we formally define ITSs and their translations, and consider certain restrictions on the generation of these translations. In Section 3 we characterize the class of translations generated by ITSs. Section 4 introduces associated ITSs (AITSs) which allow associations to be established between corresponding elements in the derivational structures of the input and output words. In Section 5 we characterize the class of translations generated by AITSs, and finally in Section 6 indicate some generalizations of these schemes.

2. Interdependent Translation Schemes

2.1. *Definitions*

We begin with some preliminary definitions.

A *word* over an alphabet V is an element of V^*, where V^* is the free monoid over V with identity λ. We let $V^+ = V^* - \{\lambda\}$.

The *length* of a word $w \in V^*$, denoted by $\lg(w)$, is the number of symbols in w, where each occurrence is counted. We define $\lg(\lambda)$ to be 0.

A *rewriting system* is a triple

$$\mathscr{R} = \langle V, \Sigma, P \rangle, \tag{2.1}$$

where V is an alphabet, $\Sigma \subseteq V$ is called the *terminal alphabet* and $P \subseteq V^+ \times V^*$ is called the set of *productions*. We will denote an element $\langle q_1, q_2 \rangle$ of P by $q_1 \to q_2$.

A word $u \in V^+$ *directly generates* a word $v \in V^*$, denoted $u \Rightarrow_{\mathscr{R}} v$, if there are words $q_1 \in V^+$ and $q_2, x, y \in V^*$ such that $u = xq_1y$, $v = xq_2y$ and $p = q_1 \to q_2$ is in P. If the above holds, we will also say that u directly generates v with *specifications* $\langle p, \lg(x) + 1 \rangle$.

A *rewriting system with axiom set* is an ordered quadruple

$$\mathscr{A} = \langle V, \Sigma, P, S \rangle, \tag{2.2}$$

where $\mathscr{R} = \langle V, \Sigma, P \rangle$ is a rewriting system and S, called the *axiom set*, is a finite subset of V.[1] We say $\mathscr{A}$ is *context-free* if each production in P is of the form $A \to q$, where $A \in V$ and $q \in V^*$. Where there is no ambiguity, we will call $\mathscr{A}$ simply a rewriting system.

Let $\Rightarrow^*_{\mathscr{R}}$ denote the reflexive transitive closure of the relation $\Rightarrow_{\mathscr{R}}$.[2] The *language generated* by $\mathscr{A}$ is defined to be

$$L(\mathscr{A}) = \{w \mid A \overset{*}{\underset{\mathscr{R}}{\Longrightarrow}} w, A \in S \text{ and } w \in \Sigma^*\}. \tag{2.3}$$

We are now in a position to define interdependent translation schemes (ITSs). An *interdependent translation scheme* is a triple

$$\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi \rangle, \tag{2.4}$$

where $\mathscr{A}_I = \langle V_I, \Sigma_I, P_I, S_I \rangle$ is a context-free rewriting system called the *syntactic structure*, $\mathscr{A}_0 = \langle V_0, \Sigma_0, P_0, S_0 \rangle$ is a context-free rewriting system called the *semantic structure* and $\Phi \subseteq P_I \times P_0$ is called the set of *translation productions*.

The subscript I indicates an *input* object, and the subscript 0 indicates an *output* object. We will similarly call the languages generated by $\mathscr{A}_I$ and $\mathscr{A}_0$ the *input language* and the *ou put language*, respectively.

Let $\mathscr{R}_I = \langle V_I, \Sigma_I, P_I \rangle$ and $\mathscr{R}_0 = \langle V_0, \Sigma_0, P_0 \rangle$. For words $u \in V_I^+$, $v \in V_I^*$, $u' \in V_0^+$ and $v' \in V_0^*$ we say $\alpha = \langle u, u' \rangle$ *directly generates* $\beta = \langle v, v' \rangle$, denoted $\alpha \Rightarrow_{\mathscr{D}} \beta$, if and only if for some $\mu = \langle p, p' \rangle \in \Phi \subseteq P_I \times P_0$

(i) for some $m \geqslant 1$

$$u \underset{\mathscr{R}_I}{\Longrightarrow} v \text{ with specifications } \langle p, m \rangle,$$

(ii) for some $m' \geqslant 1$

$$u' \underset{\mathscr{R}_0}{\Longrightarrow} v' \text{ with specifications } \langle p', m' \rangle.$$

[1] A rewriting system with axiom set is essentially a grammar [13] where we allow an initial set (the axiom set) and where terminal symbols can be rewritten.

[2] In general, if R is a relation on some set A we let R^* denote the reflexive transitive closure of R on A.

If the above holds, we also say that α directly generates β with *specifications* $\langle \mu, m, m' \rangle$. We will call the word pairs α and β *translation forms.*

The *translation generated* by $\mathscr{D}$ is then defined to be

$$T(\mathscr{D}) = \{\langle w, w' \rangle | \langle A, A' \rangle \overset{*}{\underset{\mathscr{D}}{\Longrightarrow}} \langle w, w' \rangle, A \in S_I, A' \in S_0, w \in \Sigma_I^* \text{ and } w' \in \Sigma_0^*\} \tag{2.5}$$

In the following, we will assume that the set of input productions and the set of output productions are, unless explicitly specified, the projections of the set of translation productions onto the first and second coordinates, respectively.

EXAMPLE 2.1. Consider the ITS $\mathscr{D}_1 = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi \rangle$ with syntactic structure

$$\mathscr{A}_I = \langle \{S, A, B, C, a, b, c\}, \{a, b, c\}, P_I, \{S\} \rangle$$

and semantic structure

$$\mathscr{A}_0 = \langle \{S, X, Y, d\}, \{d\}, P_0, \{S\} \rangle,$$

where the set of translation productions Φ (and hence the projection P_I onto the first coordinate and the projection P_0 onto the second coordinate) contains the following pairs[3]

1.	$S \to ABC,$	$S \to S$
2.	$A \to aA,$	$S \to X$
3.	$B \to bB,$	$X \to Y$
4.	$C \to cC,$	$Y \to dS$
5.	$A \to a,$	$S \to X$
6.	$B \to b,$	$X \to Y$
7.	$C \to c,$	$Y \to d.$

After applying production 1, productions 2, 3 and 4 can be applied in that order $(n-1)$ times, for some $n \geqslant 1$, followed by application of productions 5, 6 and 7. So we have

$$\langle S, S \rangle \Rightarrow \langle ABC, S \rangle \Rightarrow \langle aABC, X \rangle \Rightarrow \langle aAbBC, Y \rangle \Rightarrow \langle aAbBcC, dS \rangle$$
$$\Rightarrow \cdots \Rightarrow \langle a^{n-1}Ab^{n-1}Bc^{n-1}C, d^{n-1}S \rangle \Rightarrow \langle a^n b^n c^n, d^n \rangle$$

Furthermore, it is clear that the above is the only possible order of production application so that we conclude

$$T(\mathscr{D}_1) = \{\langle a^n b^n c^n, d^n \rangle \mid n \geqslant 1\}.$$

Note that the domain of $T(\mathscr{D}_1)$ is properly included in the input language. On the other hand, the range of $T(\mathscr{D}_1)$ is exactly the output language. The proper inclusion of the domain of the translation in the input language is well known both in natural languages and in programming languages, where syntactically correct sentences can be semantically invalid.

[3] The enclosing angular brackets are omitted from the pairs for readability.

2.2. *Restrictions on Translations*

The class of translations defined above is of little practical interest. We will therefore consider translations where restrictions are placed on the relation of direct generation.

For an ITS $\mathscr{D}$ (2.4), we define a binary relation $\Rightarrow_{\mathscr{D}l}$ on the set of all translation forms of $\mathscr{D}$, denoted Δ, as follows:

For any $\alpha, \beta \in \Delta$, $\alpha \Rightarrow_{\mathscr{D}l} \beta$ holds if and only if

(i) for some $\mu \in \Phi$ and positive integers m, m', $\alpha \Rightarrow_{\mathscr{D}} \beta$ holds with specifications $\langle \mu, m, m' \rangle$, and

(ii) for no $\eta \in \Phi$, $\gamma \in \Delta$ and positive integers n, n' such that $n < m$, does $\alpha \Rightarrow_{\mathscr{D}} \gamma$ hold with specifications $\langle \eta, n, n' \rangle$.

The *translation generated by* $\mathscr{D}$ *under leftmost restriction* is then defined to be

$$T_l(\mathscr{D}) = \{\langle w, w' \rangle \mid \langle A, A' \rangle \overset{*}{\underset{\mathscr{D}l}{\Longrightarrow}} \langle w, w' \rangle, A \in S_1, A' \in S_0, w \in \Sigma_1^* \text{ and } w' \in \Sigma_0^*\}. \quad (2.6)$$

We also define a binary relation $\Rightarrow_{\mathscr{D}ll}$ on Δ as follows:

For any $\alpha, \beta \in \Delta$, $\alpha \Rightarrow_{\mathscr{D}ll} \beta$ holds if and only if

(i) for some $\mu \in \Phi$ and positive integers m, m', $\alpha \Rightarrow_{\mathscr{D}l} \beta$ holds with specifications $\langle \mu, m, m' \rangle$, and

(ii) for no $\eta \in \Phi$, $\gamma \in \Delta$ and positive integer n' such that $n' < m'$, does $\alpha \Rightarrow_{\mathscr{D}l} \gamma$ hold with specifications $\langle \eta, m, n' \rangle$.

The *translation generated by* $\mathscr{D}$ *under full leftmost restriction* is then defined to be

$$T_{ll}(\mathscr{D}) = \{\langle w, w' \rangle \mid \langle A, A' \rangle \overset{*}{\underset{\mathscr{D}ll}{\Longrightarrow}} \langle w, w' \rangle, A \in S_1, A' \in S_0, w \in \Sigma_1^* \text{ and } w' \in \Sigma_0^*\}. \quad (2.7)$$

EXAMPLE 2.2. Consider the ITS $\mathscr{D}_2 = \langle \mathscr{A}_1, \mathscr{A}_0, \Phi \rangle$ with syntactic structure

$$\mathscr{A}_1 = \langle \{S, A, A', B, B', a\}, \{a\}, P_1, \{S\} \rangle$$

and semantic structure

$$\mathscr{A}_0 = \langle \{S, X, Y, Z, a\}, \{a\}, P_0, \{S\} \rangle,$$

where Φ contains

1.	$S \to A'$,	$S \to X$
2.	$A \to BB$,	$X \to X$
3.	$A' \to BB'$,	$X \to aY$
4.	$B \to A$,	$Y \to Y$
5.	$B' \to A'$,	$Y \to X$
6.	$B' \to A'$,	$Y \to Z$
7.	$A \to a$,	$Z \to Z$
8.	$A' \to a$,	$Z \to \lambda$

It is straightforward to show by induction that

$$T_l(\mathscr{D}_2) = \{\langle a^{2^n}, a^n \rangle \mid n \geqslant 1\}.$$

Informally, assume that we have a translation form $\langle A^{2^{n-1}}, a^{n-1}X\rangle$, $n \geqslant 1$, where the rightmost A is primed. Then we can apply production 2 to each unprimed A and production 3 to the primed A to obtain the translation form $\langle B^{2^n}, a^nY\rangle$, where the rightmost B is primed. Now production 4 can be applied to each unprimed B and production 5 or 6 to the primed B, giving the translation form $\langle A^{2^n}, a^nX\rangle$ or $\langle A^{2^n}, a^nZ\rangle$, respectively. In the latter case we can apply production 7 and finally production 8 to give $\langle a^{2^n}, a^n\rangle$. Thus we have

$$\langle A^{2^{n-1}}, a^{n-1}\rangle \overset{*}{\Longrightarrow} \langle A^{2^n}, a^nX\rangle \qquad \text{or} \qquad \overset{*}{\Longrightarrow} \langle a^{2^n}, a^n\rangle.$$

Production 1 gives the case for $n = 1$, and thus $\langle a^{2^n}, a^n\rangle$ is in $T(\mathscr{D}_2)$ for each $n \geqslant 1$.

We now need to show that these are the only pairs in $T(\mathscr{D}_2)$. As the output rewriting system is regular, let us say that the translation process is in *state* X (respectively Y, Z) if the output word contains the non-terminal X (respectively Y, Z). Assume that we enter state X with an input word containing k A's and l B's where the rightmost A is primed. (Note that unless some A is primed then we cannot leave state X and hence cannot reach a terminal translation form, and that primed symbols can only occur as the rightmost element of any input word.) The only applicable productions in state X are productions 2 and 3, and leftmost restriction requires that production 2 be applied to each unprimed A before production 3 can be applied to the primed A. Thus it is not possible to enter state Y without first changing each A into two B's and outputing an a. Similarly, in state Y productions 4, 5 and 6 are the only applicable productions and production 4 must be applied to each unprimed B before productions 5 or 6, so it is not possible to return to state X or to enter state Z until every B has been converted back into an A. Thus if we enter state X with k A's and l B's then, after outputing an a, we must either return to state X with $(2k + l)$ A's and no B's or enter state Z with $(2k + l)$ A's and no B's. But the only other way of entering state X is via production 1, which generates a single (primed) A. Thus we must have $l = 0$ and $k \in \{2^{n-1} \mid n \geqslant 1\}$. Furthermore, as there is no way of reaching state Z except as above, on entry to state Z we must have a translation form $\langle A^{2^n}, a^nZ\rangle$ for some $n \geqslant 1$. Once in state Z we can only change A's into a's and remove the Z so that each terminal translation form must be $\langle a^{2^n}, a^n\rangle$ for some $n \geqslant 1$. Consequently we have

$$T_l(\mathscr{D}_2) = \{\langle a^{2^n}, a^n\rangle \mid n \geqslant 1\}.$$

As leftmost restriction cannot affect the order of production application for a regular rewriting system, it is clear that we also have $T_{ll}(\mathscr{D}_2) = \{\langle a^{2^n}, a^n\rangle \mid n \geqslant 1\}$.

EXAMPLE 2.3. Similarly, the ITS $\mathscr{D}_3 = \langle \mathscr{A}_0, \mathscr{A}_1, \Phi^{-1}\rangle$, where $\mathscr{A}_1, \mathscr{A}_0$ are as above and Φ^{-1} is the converse of Φ, generates under full leftmost restriction the translation

$$T(\mathscr{D}_3) = \{\langle a^n, a^{2^n}\rangle \mid n \geqslant 1\}.$$

The proof is as above, except that input becomes output and vice versa.

3. Properties of Translations

3.1. *Characterizing Languages*

In this section we will examine the class of languages that characterize translations generated by ITSs.

We say that a language L *characterizes* [3] a translation T if there exist two homomorphisms h_I and h_0 such that $T = \{\langle h_I(w), h_0(w)\rangle \mid w \in L\}$.

We say that a language $L \subseteq (\Sigma_I \cup \Sigma_0')^*$ *strongly characterizes* a translation $T \subseteq \Sigma_I^* \times \Sigma_0{}^*$ if

(i) $\Sigma_I \cap \Sigma_0' = \varnothing$.[4]

(ii) $T = \{\langle h_I(w), h_0(w)\rangle \mid w \in L\}$, where

(a) $h_I(a) = a$ for all a in Σ_I and $h_I(a) = \lambda$ for all a in Σ_0'.

(b) $h_0(a) = \lambda$ for all a in Σ_I and h_0 is a one-to-one correspondence between Σ_0' and Σ_0.

In essence, if a language L strongly characterizes a translation T, then for a word w in L, deletion of all symbols in Σ_0' will yield an input word and deletion of all symbols in Σ_I will yield an isomorphism of the corresponding output word.

We will now introduce some lemmas leading to the theorems of this section.

LEMMA 3.1. *Every recursively enumerable language* [13] *is generated by a rewriting system* $\mathscr{A} = \langle V, \Sigma, P, \{S\}\rangle$, *where* $S \in V - \Sigma$ *and where each production in* P *is one of the following forms*

(i) $A_1 \to B_1 B_2$,

(ii) $A_1 A_2 \to B_1 B_2$,

(iii) $A_1 \to a$,

where $A_1, A_2, B_1, B_2 \in V - \Sigma$ *and* $a \in \Sigma \cup \{\lambda\}$.

The proof is straightforward (see, for example, Salomaa [13, Theorem 9.10, p. 90 and Theorem 3.1, p. 148]).

LEMMA 3.2. *If a translation* T *is strongly characterized by a recursively enumerable language then there is an ITS* $\mathscr{D}$ *such that* $T_l(\mathscr{D}) = T$.

Outline of Proof. From Lemma 3.1, we can let T be strongly characterized by a language $L(\mathscr{A})$, where $\mathscr{A} = \langle V, \Sigma, P, \{S\}\rangle$ is as in that lemma and h_I and h_0 are the two homomorphisms involved.

For each X in $V - \Sigma$, let X' and X'' be new symbols. Denote the sets of these symbols by N' and N'', respectively. Let C, D, E, F, $\bar{S}$ and U be new symbols,

[4] $\varnothing$ represents the empty set.

and let $\Sigma_I = h_I(\Sigma)$, $\Sigma_0 = h_0(\Sigma)$, $V_I = (V-\Sigma) \cup \{\bar{S}\} \cup N' \cup N'' \cup \Sigma_I$, $V_0 = \{\bar{S}, C, D, E, F, U\} \cup \Sigma_0$, and $S_I = S_0 = \{\bar{S}\}$.

Let $\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi \rangle$ be an ITS, where $\mathscr{A}_I = \langle V_I, \Sigma_I, P_I, S_I \rangle$, $\mathscr{A}_0 = \langle V_0, \Sigma_0, P_0, S_0 \rangle$ and Φ (and the projections P_I and P_0 of Φ) are defined as follows

(i) For each production $S \to B_1 B_2$ in P, where $B_1, B_2 \in V - \Sigma$, Φ contains

$$\langle \bar{S} \to B_1 B_2, \bar{S} \to C \rangle$$

and for each production $S \to a$ in P, where $a \in \Sigma \cup \{\lambda\}$, Φ contains

$$\langle \bar{S} \to h_I(a), \bar{S} \to h_0(a) \rangle.$$

(ii) For each $X \in V - \Sigma$, Φ contains

$$\langle X \to X', \quad C \to C \rangle.$$

(iii) For each production in P of the form $A_1 \to B_1 B_2$, where $A_1, B_1, B_2 \in V - \Sigma$, Φ contains

$$\langle A_1 \to B_1' B_2'', C \to E \rangle.$$

(iv) For each production in P of the form $A_1 A_2 \to B_1 B_2$, where $A_1, A_2, B_1, B_2 \in V - \Sigma$ and each $X \in V - \Sigma$ such that $X \neq A_2$, Φ contains

$$\begin{aligned} &\langle A_1 \to B_1', \quad C \to D \rangle, \\ &\langle A_2 \to B_2'', \quad D \to E \rangle, \\ &\langle X \to X, \quad D \to U \rangle, \end{aligned}$$

(v) For each $X \in V - \Sigma$, Φ contains

$$\begin{aligned} &\langle X' \to X, \quad E \to E \rangle, \\ &\langle X'' \to X, \quad E \to C \rangle. \end{aligned}$$

(vi) For each production in P of the form $A_1 \to a$, where $A_1 \in V - \Sigma$ and $a \in \Sigma \cup \{\lambda\}$, Φ contains

$$\begin{aligned} &\langle A_1 \to h_I(a), \quad C \to h_0(a)F \rangle, \\ &\langle A_1 \to h_I(a), \quad F \to h_0(a)F \rangle, \\ &\langle A_1 \to h_I(a), \quad F \to h_0(a) \rangle. \end{aligned}$$

Productions of the form (ii) allow non-terminals to be rewritten up to the intended position of application of productions (iii) or (iv). If in (iv) A_1 and A_2 are not adjacent, application of the third production introduces the non-terminal U and generation fails. Productions of the form (v) remove all primes introduced by

productions (ii)–(iv). Finally, productions of the form (vi) introduce terminal symbols in a left-to-right scan.

It is now straightforward to show that if $S \Rightarrow^*_{\mathscr{G}} w$, $w \in \Sigma^*$, then $\langle \bar{S}, \bar{S} \rangle \Rightarrow^*_{\mathscr{G}_l} \langle h_I(w), h_0(w) \rangle$, and if $\langle \bar{S}, \bar{S} \rangle \Rightarrow^*_{\mathscr{G}_l} \langle x, y \rangle$, $x \in \Sigma_I^*$ and $y \in \Sigma_0^*$, then there is some $w \in \Sigma^*$ such that $S \Rightarrow^*_{\mathscr{G}} w$, $h_I(w) = x$ and $h_0(w) = y$. Thus $T_l(\mathscr{D}) = T$.

LEMMA 3.3. *If a translation T is strongly characterized by a recursively enumerable language then there is an ITS $\mathscr{D}$ such that $T_{ll}(\mathscr{D}) = T$.*

Outline of Proof. Obvious from the construction for Lemma 3.2.

THEOREM 3.1. *A translation T is strongly characterized by a recursively enumerable language if and only if*

(i) *it is generated by an ITS uder leftmost restriction*

(ii) *it is generated by an ITS under full leftmost restriction.*

Outline of Proof. The "if" portion follows from the fact that every Turing-computable relation is strongly characterized by a recursively enumerable language. The "only if" portion follows directly from Lemmas 3.2 and 3.3.

COROLLARY 1. *A translation T is characterized by a recursively enumerable language if and only if*

(i) *it is generated by an ITS under leftmost restriction*

(ii) *it is generated by an ITS under full leftmost restriction.*

Outline of Proof. Strong characterization is a special case of characterization. Thus the "if" portion is immediate. The "only if" portion is a straightforward generalization of Lemmas 3.2 and 3.3.

COROLLARY 2. *If T is a Turing-computable relation then*

(i) *there is an ITS that generates T under leftmost restriction*

(ii) *there is an ITS that generates T under full leftmost restriction.*

The proof is obvious.

We remark that the class of languages $\mathscr{L}$ that strongly characterize translations generated without restriction by an ITS contains languages that are not context-free. This follows from Example 2.1, and the fact that the context-free languages are closed under arbitrary homomorphism. Further, it is not difficult to show that $\mathscr{L}$ is included in the class of languages generated by a matrix grammar [13]. However, it is not known how large this class is.

Similarly, the class of languages that characterize translations generated without restriction by an ITS contains non context-free languages, and is included in the class of languages generated by a matrix grammar.

3.2. *Normal Form*

THEOREM 3.2. *If T is a translation, then there exists an ITS $\mathscr{D}$ as defined in* (2.4) *such that $T = T_I(\mathscr{D})$ (respectively $T = T_{II}(\mathscr{D})$) and each translation production is of one of the forms*

(i) $\langle A_1 \to q_1,\ A_2 \to q_2\rangle$, *where* $A_1 \in V_I - \Sigma_I$, $q_1 \in (V_I - \Sigma_I)^+$, $A_2 \in V_0 - \Sigma_0$, $q_2 \in (V_0 - \Sigma_0)^+$ *and* $1 \leqslant \lg(q_i) \leqslant 2$ *for* $i = 1, 2$.

(ii) $\langle A_1 \to a_1, A_2 \to a_2\rangle$, *where* $a_1 \in \Sigma_I \cup \{\lambda\}$ *and* $a_2 \in \Sigma_0 \cup \{\lambda\}$.

Outline of Proof. Obvious from the construction for Lemma 3.2 (respectively Lemma 3.3) introducing non-terminal stand-ins in the usual manner.

3.3. *Length Increasing Translation Schemes*

We say an ITS is *length increasing* if and only if every translation production $\langle A_1 \to q_1, A_2 \to q_2\rangle$ satisfies $\lg(q_1 q_2) \geqslant 2$, with the possible exception of those productions where A_1 and A_2 are, respectively, input and output axioms and A_1 does not appear on the right-hand side of any input production and A_2 does not appear on the right-hand side of any output production.

LEMMA 3.4. *Every context-sensitive language is generated by a rewriting system $\mathscr{A}$ as given in Lemma* 3.1, *and where each production is length increasing* [13], *except possibly for the production $S \to \lambda$, in which case S does not appear on the right-hand side of any other production in $\mathscr{A}$.*

Outline of Proof. As in Salomaa [13].

LEMMA 3.5. *If a translation T is strongly characterized by a context-sensitive language, then there exists a length increasing ITS $\bar{\mathscr{D}}$ such that $T = T_I(\bar{\mathscr{D}})$.*

Outline of Proof. From Lemma 3.4, we can let T be characterized by the context-sensitive language $L(\mathscr{A})$, where $\mathscr{A}$ is as in that lemma. Using the same construction as in the proof of Lemma 3.1, we construct the ITS $\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi\rangle$. From Lemma 3.4, we also have that $\mathscr{D}$ is length increasing, except for productions of the form $\langle A \to a, F \to b\rangle$, where $A \in V_I$, and either $a = \lambda$ and $b \in \Sigma_0$ or $a \in \Sigma_I$ and $b = \lambda$. We now construct an ITS $\bar{\mathscr{D}}$ which generates T and in which these productions do not appear.

Let $\bar{V}_I = V_I \cup ((V_I - \Sigma_I) \times (V_I - \Sigma_I))$. Let $\bar{\mathscr{D}} = \langle \bar{\mathscr{A}}_I, \bar{\mathscr{A}}_0, \bar{\Phi}\rangle$ be an ITS, where $\bar{\mathscr{A}}_I = \langle \bar{V}_I, \Sigma_I, \bar{P}_I, S_I\rangle$, $\bar{\mathscr{A}}_0 = \langle V_0, \Sigma_0, \bar{P}_0, S_0\rangle$ and $\bar{\Phi}$ is defined as follows:

(i) For each production $\langle \bar{S} \to B_1 B_2, \bar{S} \to C\rangle$ in Φ, where $B_1, B_2 \in V_I - \Sigma_I$, $\bar{\Phi}$ contains

$$\langle \bar{S} \to [B_1, B_2], \bar{S} \to C\rangle$$

and for each production $\langle \bar{S} \to a_1, \bar{S} \to a_2\rangle$ in Φ, where $a_1 \in \Sigma_I \cup \{\lambda\}$, $a_2 \in \Sigma_0 \cup \{\lambda\}$, $\bar{\Phi}$ contains

$$\langle \bar{S} \to a_1, \bar{S} \to a_2\rangle.$$

(ii) For each production in Φ of the form $\langle A \to B_1 B_2, C \to E\rangle$, where $A, B_1, B_2 \in V_I - \Sigma_I$, and each $X \in V_I - \Sigma_I$, $\bar{\Phi}$ contains

$$\langle A \to B_1 B_2, C \to E\rangle,$$
$$\langle [A, X] \to B_1 [B_2, X], C \to E\rangle,$$
$$\langle [X, A] \to X[B_1, B_2], C \to E\rangle.$$

(iii) For each production Φ of the form $\langle A \to B, \ Y \to Z\rangle$, where $A, B \in V_I - \Sigma_I$, $Y, Z \in V_0 - \Sigma_0$, and each $X \in V_I - \Sigma_I$, $\bar{\Phi}$ contains

$$\langle A \to B, Y \to Z\rangle,$$
$$\langle [A, X] \to [B, X], Y \to Z\rangle,$$
$$\langle [X, A] \to [X, B], Y \to Z\rangle.$$

(iv) For each production in Φ of the form $\langle A \to a, Y \to bF\rangle$, where $A \in V_I - \Sigma_I$, $a \in \Sigma_I \cup \{\lambda\}$, $Y \in \{F, C\}$, and $b \in \Sigma_0 \cup \{\lambda\}$, $\bar{\Phi}$ also contains

$$\langle A \to a, Y \to bF\rangle.$$

(v) For each pair of productions $\langle A_1 \to a_1, Y \to b_1 F\rangle$ and $\langle A_2 \to a_2, F \to b_2\rangle$ in Φ, where $A_1, A_2 \in V_I - \Sigma_I$, $a_1, a_2 \in \Sigma_I \cup \{\lambda\}$, $Y \in \{F, C\}$ and $b_1, b_2 \in \Sigma_0 \cup \{\lambda\}$, $\bar{\Phi}$ contains

$$\langle [A_1, A_2] \to a_1 a_2, Y \to b_1 b_2\rangle.$$

The construction simply combines the last input symbol with the one to its left. Clearly, $T_l(\bar{\mathscr{D}}) = T_l(\mathscr{D})$. Further, for productions of the form (iv) exactly one of a and b is empty, and for productions of the form (v) exactly one of a_i and b_i for $i = 1, 2$, is empty. Thus $\bar{\mathscr{D}}$ is length increasing.

In order to prove the next lemma (and some subsequent lemmas), we introduce the notion of a matrix grammar.

A (*context-free*) *matrix grammar* [13] is an ordered quadruple

$$G = \langle N, \Sigma, M, S\rangle, \tag{3.1}$$

where

N is an alphabet called the *nonterminal alphabet*,

Σ is an alphabet called the *terminal alphabet*, $N \cap \Sigma = \varnothing$,

$S \in N$ is called the *initial symbol*,

M is a finite set of finite nonempty sequences where elements are ordered pairs $\langle A, q\rangle$, where $A \in N$ and $q \in (N \cup \Sigma)^*$.

The pairs are referred to as *productions* and written $A \to q$.

The sequences are referred to as *matrices* and written

$$m = [A_1 \to q_1, \ldots, A_n \to q_n], n \geqslant 1.$$

Let $V = N \cup \Sigma$.

For a matrix grammar (3.1), we define a binary relation $\Rightarrow_G$ on the set V^* as follows. For any $u, v \in V^*$, $u \Rightarrow_G v$ holds if there exist an integer $n \geqslant 1$ and words

$$w_1, \ldots, w_{n+1}; A_1, \ldots, A_n; q_1, \ldots, q_n; x_1, \ldots, x_n; y_1, \ldots, y_n$$

over V such that

(i) $w_1 = u$ and $w_{n+1} = v$,

(ii) $m = [A_1 \to q_1, \ldots, A_n \to q_n] \in M$,

(iii) $w_i = x_i A_i y_i$ and $w_{i+1} = x_i q_i y_i$, $i = 1, \ldots, n$.

If the above holds, we also say that $x \Rightarrow_G y$ with *specifications* $\langle m, \lg(x_1) + 1 \rangle$. The language *generated* by the matrix grammar G is defined to be

$$L(G) = \{w \mid S \overset{*}{\underset{G}{\Longrightarrow}} w, w \in \Sigma^*\}.$$

We now define leftmost restriction on the relation of direct generation for matrix grammars.

Let $G = \langle N, \Sigma, M, S \rangle$ be a matrix grammar as defined in (3.1). A binary relation $\Rightarrow_{Gl}$ on V^* is defined as follows. For any $u, v \in V^*$, $u \Rightarrow_{Gl} v$ holds if and only if

(i) for some $m \in M$ and positive integer k, $u \Rightarrow_G v$ with specifications $\langle m, k \rangle$ and

(ii) for no $m' \in M$, $v' \in V^*$, and positive integer k' such that $k' < k$, does $u \Rightarrow_G v'$ hold with specifications $\langle m', k' \rangle$.

The language *generated* by the matrix grammar G *under leftmost restriction* is defined to be

$$L_l(G) = \{w \mid S \overset{*}{\underset{Gl}{\Longrightarrow}} w, w \in \Sigma^*\}.$$

We will say that a matrix grammar $G = \langle N, \Sigma, M, S \rangle$ is *length-increasing* if for all productions $[A_1 \to q_1, \ldots, A_n \to q_n]$ in M, $\lg(q_1 \cdots q_n) \geqslant n$. We then have the following lemma.

LEMMA 3.6. *If a language L is generated under leftmost restriction by a length-increasing matrix grammar then L is context sensitive.*

The proof is a straightforward generalization of that given for λ-free matrix grammars in Salomaa [13, Theorem 3.2, p. 148].

LEMMA 3.7. *If T is generated under leftmost restriction by a length increasing ITS $\mathscr{D}$ then T is strongly characterized by a context-sensitive language.*

Outline of Proof. Let $\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi \rangle$ as defined in (2.4) and without loss of generality assume that terminal symbols cannot be rewritten. Define an isomorphism h of V_0^* such that for each X in V_0, $h(X) = \bar{X}$, where $\bar{X}$ is a new symbol not in V_I. Let S be a new symbol not in $V_I \cup h(V_0)$. Let $G = \langle N, \Sigma, M, S \rangle$ be a matrix grammar where $\Sigma = \Sigma_I \cup h(\Sigma_0)$, $N = (V_I - \Sigma_I) \cup h(V_0 - \Sigma_0) \cup \{S\}$, and M contains the following matrices:

(i) For each $X \in S_I$ and $Y \in S_0$, M contains $[S \to Xh(Y)]$.

(ii) For each production in Φ of the form $\langle A_1 \to q_1, A_2 \to q_2 \rangle$, M contains

$$[A_1 \to q_1, h(A_2) \to h(q_2)].$$

Clearly, $T(\mathscr{D})$ is strongly characterized by $L_t(G)$. Note that in productions of the form (ii), $\lg(q_1 h(q_2)) \geqslant 2$ but either q_1 or $h(q_2)$ may be empty. Thus from Lemma 3.6 we have that $T(\mathscr{D})$ is strongly characterized by a context-sensitive language.

THEOREM 3.3. *A translation T is strongly characterized by a context-sensitive language if and only if it is generated under leftmost restriction by a length increasing ITS.*

The proof follows directly from Lemmas 3.5 and 3.7.

The same result holds for full leftmost restriction. We leave the proof as a (relatively uninteresting) exercise for the reader.

4. TRANSLATION SCHEMES WITH ASSOCIATIONS

4.1. *Definitions*

For many applications, particularly in natural language processing, it is preferable to consider the mapping specified by a translation scheme to be on trees rather than strings (e.g., [5]). Furthermore, one often wishes to associate subtrees of the input tree with subtrees of the output tree while at the same time retaining the possibility of contextual dependence. We introduce such translation schemes below, after some preliminary definitions.

Let Z^+ be the set of positive integers, and Z_+^* be the set of finite strings of positive integers. A *value tree* (or simply, *tree*) on an alphabet V is a function from a closed subset of Z_+^* into V, where $E \subseteq Z_+^*$ is closed if and only if

(i) for $w, u, v \in Z_+^*$, $w \in E$ and $w = uv$ implies $u \in E$;

(ii) for $w \in Z_+^*$ and $m, n \in Z_+$, $wn \in E$ and $m \leqslant n$ implies $wm \in E$.

A value tree clearly corresponds to an ordered rooted tree (representing the domain of the function) where the nodes are labelled with elements of V (representing the range of the function).

FIGURE 1

Given a tree t on an alphabet V, the *subtree* of t *rooted at* $i \in \mathrm{dom}(t)$,[5] denoted t_i, is defined to be the restriction of t to the set $\{y \mid y \in \mathrm{dom}(t) \text{ and } y = iz \text{ for some } z \in Z_+^*\}$.

We denote by $\mathrm{Fr}(t)$ the union of all subtrees consisting of one element.

The yield of a tree t on V, denoted yd_t, is a map from the set of subtrees of t into V^*. The definition is inductive.

$$yd_t(t_i) = t_i(i) \qquad \text{for} \quad t_i \in \mathrm{Fr}(t)$$
$$= yd_t(t_{i1})\, yd_t(t_{i2}) \cdots yd_t(t_{im})$$
$$\text{for } m \geqslant 1 \text{ such that } im \in \mathrm{dom}(t) \text{ and } i(m+1) \notin \mathrm{dom}(t).$$

Where no confusion can arise, $yd_t(t)$ will simply be denoted by $yd(t)$. Thus $yd(t)$ is the string of symbols labelling (in order) the leaf nodes of t.

EXAMPLE 4.1. The value tree t on $\{a, b, c\}$ given by $t = \{\langle \lambda, a\rangle\langle 1, b\rangle\langle 2, b\rangle\langle 11, c\rangle\langle 12, b\rangle\}$ is represented by the rooted tree of Fig. 1. The subtree $t_1 = \{\langle 1, b\rangle\langle 11, c\rangle\langle 12, b\rangle\}$, $\mathrm{Fr}(t) = \{\langle 11, c\rangle\langle 12, b\rangle\langle 2, b\rangle\}$, and $yd(t) = cbb$.

We will also need the concept of association of value trees.

An *association* ψ *of trees* t, t' on V and V', respectively, is a partial one-to-one map from $\mathrm{dom}(t)$ into $\mathrm{dom}(t')$. For i in $\mathrm{dom}(\psi)$, we say that node i in t is *associated with node* $\psi(i)$ in t'. If i is not in $\mathrm{dom}(\psi)$, we say that node i in t is *unassociated*. If $i \in \mathrm{dom}(t)$ and $j \in \mathrm{dom}(t')$ are associated, then we say that the subtrees t_i and t'_j are associated.

We will also require that ψ be such that if subtrees t_i and t'_j are associated, and subtrees t_k and t'_l are also associated, then $k \neq i$ and $k \in \mathrm{dom}(t_i)$ implies $j \notin \mathrm{dom}(t'_l)$. That is, t_i cannot properly contain a subtree that is associated with a subtree containing t'_j.

Assume that the root nodes of t and t' are associated. Then for a pair of nodes $i \in \mathrm{dom}(t)$ and $j \in \mathrm{dom}(t')$ consider the set of subtrees of t that contain i and are associated with some subtree containing j. This set can be naturally ordered by set inclusion and we will denote the smallest element (subtree) by $C_\psi(i, j)$. Then for $i \in \mathrm{dom}(t)$ and j, $k \in \mathrm{dom}(t')$, we say that node i is *more closely associated* with node j than with node k if and only if $C_\psi(i, j) \subset C_\psi(i, k)$.

[5] For a function f, $\mathrm{dom}(f)$ is the domain of f and $\mathrm{ran}(f)$ is the range of f.

EXAMPLE 4.2. For the trees t and t' of Fig. 1a, b, respectively, let the association ψ be given by $\psi = \{\langle\lambda, \lambda\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 11, 21\rangle\}$. Then $C_\psi(11, 21) = \langle t_{11}, t'_{21}\rangle$, $C_\psi(11, 22) = \langle t_1, t'_2\rangle$, and $C_\psi(11, 21) \subset C_\psi(11, 22)$.

We now define the relation of direct generation on value trees rather than on words. These value trees correspond to the derivation trees of a context-free grammar.

Given a rewriting system $\mathscr{R} = \langle V, \Sigma, P\rangle$ we say a tree u on V *directly generates* a tree v on V, denoted $u \Rightarrow_{\mathscr{R}} v$, if for some $n \in Z_+^*$ and some A, $q_i \in V$, $1 \leqslant i \leqslant m$, $\langle n, A\rangle$ is in $\mathrm{Fr}(u)$, $p = A \to q_1 \cdots q_m$ is in P and $v = u \cup \{\langle ni, q_i\rangle \mid 1 \leqslant i \leqslant m\}$.

If the above holds, we also say that u directly generates v with *specifications* $\langle p, n\rangle$.

We can now generalize the definition of direct generation for an ITS to include trees by letting u, v be trees on V_I, u', v' be trees on V_0 and $m, m' \in Z_+^*$ in that definition (Section 2.1).

At this stage we are ready to introduce the concept of an associated interdependent translation scheme (AITS).

An *associated interdependent translation scheme*, $\mathscr{D}$, is a triple

$$\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi\rangle, \tag{4.1}$$

where $\mathscr{A}_I$ and $\mathscr{A}_0$ are as defined by (2.4) and each element of Φ is a triple $\langle A \to q, A' \to q', \phi\rangle$, where $A \to q \in P_I$, $A' \to q' \in P_0$ and ϕ is a partial map that associates with some symbols occurring in q a symbol occurring in q'. More formally, ϕ is a partial one-to-one map from $\{1, 2, \ldots, \lg(q)\}$ into $\{1, 2, \ldots, \lg(q')\}$, where for i in $\mathrm{dom}(\phi)$, the ith symbol (from the left) in q is said to be *associated with* the $\phi(i)$th symbol (from the left) in q'.

Let $\bar{\Phi}$ be the projection of Φ onto $P_I \times P_0$ and let $h: \Phi \to \bar{\Phi}$ denote the corresponding map. Let $\bar{\mathscr{D}}$ denote the ITS $\langle \mathscr{A}_I, \mathscr{A}_0, \bar{\Phi}\rangle$, and let u, v be trees on V_I and u', v' be trees on V_0. Let χ be an association of u and u' and let ψ be an association of v and v'. Then we say $\alpha = \langle u, u', \chi\rangle$ *directly generates* $\beta = \langle v, v', \psi\rangle$, denoted $\alpha \Rightarrow_{\mathscr{D}} \beta$, if and only if for some $\mu = \langle p, p', \phi\rangle \in \Phi$

(i) for some $m, m' \in Z_+^*$

$$\langle u, u'\rangle \underset{\bar{\mathscr{D}}}{\Longrightarrow} \langle v, v'\rangle \text{ with specifications } \langle h(\mu), m, m'\rangle$$

(ii) for no $v \in \Phi$, w, w' trees on V_I and V_0, respectively, and $n' \in Z_+^*$ such that $C\chi(m, n') \subset C\chi(m, m')$ does

$$\langle u, u'\rangle \underset{\bar{\mathscr{D}}}{\Longrightarrow} \langle w, w'\rangle \text{ with specifications } \langle h(v), m, n'\rangle$$

and

(iii) $\psi = \chi \cup \{\langle mi, m'j\rangle \mid \langle i, j\rangle \in \phi\}$.

Condition (i) states that the translation production μ is *applicable* at nodes m and m' if it is applicable at these nodes in the corresponding ITS $\bar{\mathscr{D}}$. Condition (ii) simply requires that of all the applicable nodes in u', one of those most closely associated

with node m is to be used— it can be interpreted as a scope rule for production application. Condition (iii) defines the association of the nodes in v and v' to be the same as the association of nodes in u and u', except for the additional nodes which are associated as in the translation production.

The *translation generated* by $\mathscr{D}$ is then defined to be

$$T(\mathscr{D}) = \{\langle yd(u), yd(u')\rangle \mid \iota \overset{*}{\underset{\mathscr{D}}{\Longrightarrow}} \langle u, u', \phi\rangle \text{ for some association } \phi \text{ and } \iota \in I_{\mathscr{D}}, yd(u) \in \Sigma_I^* \text{ and } yd(u') \in \Sigma_0^*\}, \tag{4.2}$$

where $I_{\mathscr{D}} = \{\langle\langle\lambda, A\rangle, \langle\lambda, A'\rangle\langle\lambda, \lambda\rangle\rangle \mid A \in S_I \text{ and } A' \in S_0\}$.

EXAMPLE 4.3. Consider the AITS $\mathscr{D}_{A1} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi\rangle$ with

$$\mathscr{A}_I = \langle\{S, X, Y, Z, a\}, \{a\}, P_I, \{S\}\rangle,$$
$$\mathscr{A}_0 = \langle\{S, A, B, C, a\}, \{a\}, P_0, \{S\}\rangle,$$

where Φ (and the projections P_I and P_0 onto the first and second coordinates, respectively) contains[6]

1.	$S \to X^1$,	$S \to AC$
2.	$X \to X$,	$A \to BB$
3.	$X \to Ya$,	$C \to C$
4.	$Y \to Y$,	$B \to A$
5.	$Y \to X$,	$C \to C$
6.	$Y \to Z$,	$C \to C$
7.	$Z \to Z$,	$A \to a$
8.	$Z \to \lambda$,	$C \to \lambda$

Then the associated translation generated by $\mathscr{D}_{A1}$ is

$$T(\mathscr{D}_{A1}) = \{\langle a^n, a^{2^n}\rangle \mid n \geqslant 1\}.$$

The proof of this follows essentially the same lines as the proof outlined in Examples 2.2 and 2.3, i.e., in state X production 3 cannot be applied until each A has been converted into two B's and in state Y productions 5 or 6 cannot be applied until each B has been converted back into an A. Whereas leftmost restriction ensured this in Examples 2.2 and 2.3, it is restriction (ii) that ensures it in the present case, i.e., X, Y, Z are less closely associated with C than they are with each A and B, and hence productions 2, 4 and 7 always have priority over productions 3, 5, 6 and 8.

For example, consider the following derivation, where we use the notation $\langle p, \alpha, \beta\rangle$ as applying production p to node α of the input tree and node β of the output tree:

After applying production 1, input node 1 is more closely associated with output

[6] Associations are represented by superscripts.

node 1 than with output node 2, and thus under condition (ii) of the preceding definition production 2 must be used in preference to production 3. Thus we have

$$\langle 1, \lambda, \lambda \rangle \Rightarrow \langle 2, 1, 1 \rangle \Rightarrow \langle 3, 1.1, 2 \rangle.$$

At this stage, input node 1.1.1 is more closely associated with output nodes 1.1 and 1.2 than with output node 2.1, so whereas we can apply production 4 at output node 1.1 or 1.2, we cannot apply productions 5 or 6 at output node 2.1. So continuing we have

$$\langle 3, 1.1, 2 \rangle \Rightarrow \langle 4, 1.1.1, 1.1 \rangle \Rightarrow \langle 4, 1.1.1.1, 1.2 \rangle.$$

Only productions 5 and 6 are now applicable, so we can continue

$$\langle 4, 1.1.1.1, 1.2 \rangle \Rightarrow \langle 6, 1.1.1.1.1, 2.1 \rangle \Rightarrow \langle 7, 1.1.1.1.1.1, 1.1.1 \rangle$$

$$\langle 7, 1.1.1.1.1.1.1, 1.2.1 \rangle \Rightarrow \langle 8, 1.1.1.1.1.1.1.1, 2.1.1 \rangle,$$

thus giving the translation element $\langle a, a^2 \rangle$.

Simple and practical examples of associated translation schemes are difficult to find, firstly because no restrictions are placed on the generation of translation forms and secondly, because output in limited to being string valued. We give a variation on a familiar example below.

EXAMPLE 4.4. Consider the AITS $\mathscr{D}_{A2} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi \rangle$, where

$$\mathscr{A}_I = \langle \{S, D, E, \bar{a}, \bar{b}, \bar{c}, (,), a, b, c, +, \times, \cdot\}, \{(,), a, b, c, +, \times, \cdot\}, P_I, \{S\} \rangle,$$

$$\mathscr{A}_0 = \langle \{S, D, +, \times, I, \bar{a}, \bar{b}, \bar{c}, a, b, c\}, \{\bar{a}, \bar{b}, \bar{c}, a, b, c\}, P_0, \{S\} \rangle,$$

where Φ contains

1.	$S \to D \cdot E,$	$S \to DP$
2.	$E \to E^1 + E^2,$	$+ \to TP$
3.	$E \to E^1 \times E^2,$	$\times \to IT$
4.	$E \to (E^1),$	$I \to P$
5.	$E \to \bar{x},$	$\bar{x} \to \bar{x}$
6.	$\bar{x} \to x,$	$I \to x$
7.	$D \to xD,$	$D \to \bar{x}D$
8.	$D \to \lambda,$	$D \to \lambda$

for $P = +, \times$, and I; $T = \times$ and I; and $x = a, b$ and c.

In the above scheme, the non-terminal E can only generate an a (respectively b, c) if $\bar{a}$ (respectively $\bar{b}, \bar{c}$) already occurs in the output tree, and this can only happen if a (respectively b, c) occurs in the input subtree rooted at D. Otherwise we simply have the usual rewriting system for infix expressions, except that the information specifying precedence of operators is contained in the semantic structure.

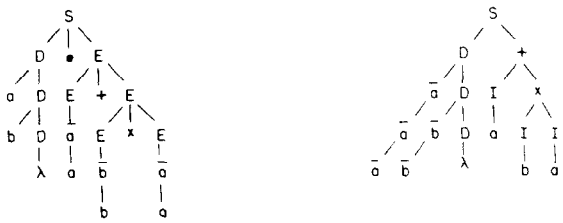

FIGURE 2

Interpreting symbols before the " · " as declared variables, the domain of the translation thus contains words of the form ⟨declarations⟩ · ⟨infix expression⟩ where only those variables appearing in ⟨declarations⟩ can appear in ⟨infix expression⟩. An imaginative reader could interpret the output as a symbol table (D) together with an expression tree representing ⟨infix expression⟩. Production 7 generates the symbol table, and production 5 checks for the existence of the given variable in the symbol table. I is taken to be the identity function.

For example, the underlying tree structures for the translation element $\langle ab \cdot a + b \times a, \bar{a}\bar{b}aba \rangle$ are given in Fig. 2.

4.2. *Leftmost Restriction and Full Leftmost Restriction*

Leftmost restriction and full leftmost restriction can be defined for IATSs as for ITSs. As an ITS is a special case of an AITS it immediately follows from Theorem 1 that the class of translations generated by AITS under leftmost restriction (respectively, full leftmost restriction) is equal to that generated by ITSs under leftmost restriction (respectively, full leftmost restriction). We will not, therefore, further consider leftmost restriction or full leftmost restriction on AITSs, although clearly they are of interest from a practical or constructive point of view.

5. PROPERTIES OF ASSOCIATED TRANSLATIONS

5.1. *Characterizing Languages*

In this section we consider the characterizing languages of translations generated by AITSs. To do this, we introduce a different form of restriction on the relation of direct generation for matrix grammars.

We define the application of a matrix in the following *appearance checking* sense: If the left side of a production is not a subword of the word under scan, then we may move onto the next production. In the subsequent formal definition, a subset P_1 of the set P of productions is specified such that appearance checking is possible only for productions in P_1.

Let $G = \langle N, \Sigma, M, S \rangle$ be a matrix grammar as defined by (3.1). Let P be the set of occurrences of productions in the matrices M, and P_1 a subset of P. A binary relation $\Rightarrow_{ac}$ (which is to be understood as being indexed by G and P_1) on the set V^* is defined as follows:

For any $u, v \in V^*$, $u \Rightarrow_{ac} v$ holds if there exists an integer $n \geqslant 1$ and words

$$w_1,..., w_{n+1}; A_1,..., A_n; q_1,..., q_n; x_1,..., x_n; y_1,..., y_n$$

over V such that

(i) $w_1 = u$ and $w_{n+1} = v$,

(ii) $m = [A_1 \to q_1,..., A_n \to q_n] \in M$,

(iii) for every $i = 1,..., n$ either $w_i = x_i A_i y_i$ and $w_{i+1} = x_i q_i y_i$ or else the (occurrence of the) production $A_i \to q_i$ belongs to P_1, A_i is not a subword of w_i, and $w_i = w_{i+1}$.

Let $\Rightarrow_{ac}^*$ be the reflexive transitive closure of $\Rightarrow_{ac}$. The language generated by G with *appearance checking* for productions in P_1 is defined by

$$L_{ac}(G, P_1) = \{w \mid S \overset{*}{\underset{ac}{\Longrightarrow}} w, w \in \Sigma^*\}.$$

LEMMA 5.1. *Any language generated with appearance checking by a matrix grammar is generated by a matrix grammar* $G = \langle N, \Sigma, M, S\rangle$, *where* $[A_1 \to q_1, A_2 \to q_2,..., A_n \to q_n]$ *in* M *implies* $n = 2$ *and* $0 \leqslant \lg(q_i) \leqslant 2$ *for* $i = 1,..., n$ *and where appearance checking is restricted to the first production of each matrix.*

Outline of Proof. As in Salomaa [13], using standard techniques to make the length of the right-hand side of each production $\leqslant 2$.

LEMMA 5.2. *A language is recursively enumerable if and only if it is generated with appearance checking by a matrix grammar.*

Outline of Proof. As in Salomaa [13].

LEMMA 5.3. *If a translation* T *is strongly characterized by a language generated by a matrix grammar with appearance checking then there is an AITS* $\mathscr{D}$ *such that* $T(\mathscr{D}) = T$.

Outline of Proof. From Lemma 5.1 we can let T be strongly characterized by a language $L_{ac}(G, P_1)$, where $G = \{N, \Sigma, M, S\}$ is as in that lemma, P_1 only contains productions that occur as the first production of some matrix in M, and where h_I and h_0 are the two homomorphisms involved. Extend h_I and h_0 to N such that $h_I(X) = h_0(X) = X$ for each X in N.

Let the number of matrices in G be n. Number the matrices from 1 to n, and let $[A_{i1} \to x_{i1} y_{i1}, A_{i2} \to x_{i2} y_{i2}]$ be the ith matrix, where $A_{i1}, A_{i2} \in N$, and x_{i1}, y_{i1}, x_{i2} and $y_{i2} \in N \cup \Sigma \cup \{\lambda\}$. Let $B, C, D, \bar{S}$ and U be new symbols, and let $W = \{[i, j] \mid 1 \leqslant i \leqslant n, \ 1 \leqslant j \leqslant 4\}$. Let $\Sigma_I = h_I(\Sigma)$, $\Sigma_0 = h_0(\Sigma)$, $V_I = N \cup \{\bar{S}\} \cup W \cup \Sigma_I$, $V_0 = N \cup \{B, C, D, \bar{S}, U\} \cup W \cup \Sigma_0$ and $S_I = S_0 = \{\bar{S}\}$.

Let $\mathscr{D} = \langle \mathscr{A}_I, \mathscr{A}_0, \Phi\rangle$ be an AITS, where $\mathscr{A}_I = \langle V_I, \Sigma_I, P_I, S_I\rangle$, $\mathscr{A}_0 = \langle V_0, \Sigma_0, P_0, S_0\rangle$ and where Φ (and the projections P_I and P_0 of Φ) are defined as follows:

(i) Φ contains

$$\langle \bar{S} \to S, \qquad \bar{S} \to SB, \qquad \{\langle 1, 1\rangle\}\rangle.$$

(ii) For $i = 1,..., n$, Φ contains

$$\begin{array}{lll}
\langle A_{i1} \to [i, 1], & B \to C, & \varnothing\rangle, \\
\langle [i, 1] \to h_I(x_{i1}\, y_{i1}), & A_{i1} \to h_0(x_{i1})[i, 1], & \psi_1\rangle, \\
\langle A_{i2} \to [i, 2], & [i, 1] \to h_0(y_{i1}), & \varnothing\rangle, \\
\langle [i, 2] \to h_I(x_{i2})[i, 3], & A_{i2} \to h_0(x_{i2}\, y_{i2}), & \psi_2\rangle, \\
\langle [i, 3] \to h_I(y_{i2}), & C \to B, & \varnothing\rangle,
\end{array}$$

where for $j = 1, 2$

$$\begin{array}{ll}
\psi_j = \varnothing & \text{if } x_{ij} \text{ or } y_{ij} \text{ are in } \Sigma \cup \{\lambda\} \\
= \{\langle 1, 1\rangle, \langle 2, 2\rangle\} & \text{otherwise.}
\end{array}$$

(iii) For $1 \leqslant i \leqslant n$ such that the first production of matrix i is in P_1, Φ contains

$$\begin{array}{lll}
\langle A_{i2} \to [i, 4], & B \to D, & \varnothing\rangle, \\
\langle [i, 4] \to [i, 2], & A_{i1} \to U, & \varnothing\rangle, \\
\langle [i, 4] \to [i, 2], & D \to C, & \varnothing\rangle.
\end{array}$$

(iv) For $i = 1,..., n$, Φ contains

$$\langle [i, 3] \to h_I(y_{i2}), \quad C \to \lambda, \quad \varnothing\rangle.$$

Note that if the first production in (iii) is applied then the translation will fail if the non-terminal symbol A_{i1} appears anywhere in the output word (and consequently the input word) of the current translation form.

Thus productions of the form (iii) simulate appearance checking in the matrix grammar G. Productions of the form (iv) terminate the translation.

Let $\iota = \langle\langle \lambda, \bar{S}\rangle, \langle \lambda, \bar{S}\rangle, \langle \lambda, \lambda\rangle\rangle$. Clearly, if $S \Rightarrow^*_{Gac} w$, $w \in \Sigma^*$ then $\langle \bar{S}, \bar{S}, \iota\rangle \Rightarrow^*_{\mathscr{D}} \langle h_I(w), h_0(w), \varnothing\rangle$, and if $\langle \bar{S}, \bar{S}, \iota\rangle \Rightarrow^*_{\mathscr{D}} \langle x, y, \phi\rangle$ for some association ϕ, $x \in \Sigma_I^*$, and $y \in \Sigma_0^*$, then there is some $w \in \Sigma^*$ such that $S \Rightarrow^*_{Gac} w$, $h_I(w) = x$ and $h_0(w) = y$. Thus $T(\mathscr{D}) = T$.

THEOREM 5.1. *A translation T is strongly characterized by a recursively enumerable language if and only if it is generated by an AITS.*

Outline of Proof. The "if" portion is as for Theorem 3.1. The "only if" portion follows directly from Lemma 5.3.

COROLLARY 1. *A translation T is characterized by a recurcively enumerable language if and only if it is generated by an AITS.*

Outline of Proof. Strong characterization is special case of characterization. Thus the "if" portion is immediate. The "only if" portion is a straightforward generalization of Lemma 5.3.

COROLLARY 2. *If T is a Turing-computable relation then the translation represented by T is generated by an AITS.*

The proof is obvious.

5.2. *Normal Form*

THEOREM 5.2. *If T is a translation, then there exists an AITS $\mathscr{D}$ as defined in* (4.1) *such that $T = T(\mathscr{D})$ and each production in Φ is of one of the forms*

(i) $\langle A_1 \to q_1, A_2 \to q_2, \phi \rangle$ *where* $A_1 \in V_I - \Sigma_I$, $q_1 \in (V_I - \Sigma_I)^+$, $A_2 \in V_0 - \Sigma_0$, $q_2 \in (V_0 - \Sigma_0)^+$, $1 \leqslant \lg(q_i) \leqslant 2$ *for* $i = 1, 2$ *and* ϕ *is a partial map*: $\{1, 2\} \to \{1, 2\}$.

(ii) $\langle A_1 \to a_1, A_2 \to a_2, \varnothing \rangle$ *where* $a_1 \in \Sigma_I \cup \{\lambda\}$ *and* $a_2 \in \Sigma_0 \cup \{\lambda\}$.

The proof is obvious from the construction for Lemma 5.3, introducing non-terminal stand-ins where necessary.

6. CONCLUSIONS

One of the characteristic properties of many programming languages and most natural languages is that their semantic interpretation shows some form of contextual dependence. Secondly, the structure of the semantic domain is rarely similar to that of the syntactic domain. Interdependent translation schemes (ITSs and AITSs) differ from previous formalisms for translations largely in that such mappings can be readily specified. Furthermore, it is shown that ITSs under leftmost restriction and AITSs have the power of Turing machines, which is desirable at least for natural language applications.

Under some relatively straightforward generalizations, AITSs can be shown to encompass most of the previously defined translation schemes. As defined herein, AITSs clearly include the syntax directed translation schemes of Lewis and Stearns [10] and Aho and Ullman [2, 3] as special cases. By allowing each input symbol occurring in a translation production to be associated with a set of output symbols, and then by rewriting in parallel associated subtrees appearing in the translation forms, the generalized syntax directed translation scheme of Aho and Ullman [1] also becomes a special case.

Another possibility is to generalize the notion of direct generation for rewriting systems. Each production in a rewriting system can clearly be represented as a value tree of height 2. The definition of direct generation in Section 4.1 can then be reformulated to read $u \Rightarrow_{\mathscr{R}} v$ if for some $n \in Z_+^*$ and some $A \in V$, $\langle n, A \rangle$ is in $\mathrm{Fr}(u)$, $p \in P$ has root node λ labelled A and $v = u \cup \{\langle ni, p(i) \rangle \mid i \in \mathrm{dom}(p)\}$. Under this definition, we can relax the restriction on productions to be of height 2, allowing

value trees of arbitrary height. The generalization follows through to be the definition of direct generation in an AITS. The resulting translation scheme, together with the generalization to multiply associated subtrees, would then include tree automata with output as defined by Rounds [12] and Thatcher [14, 15] as special cases. Of course, none of these generalizations increase the power of AITSs, but they may improve constructibility.

REFERENCES

1. A. V. AHO AND J. D. ULLMAN, The theory of languages, *Math. Systems Theory* 2:2 (1968), 97–126.
2. A. V. AHO AND J. D. ULLMAN, Syntax directed translations and the pushdown assembler, *J. Comput. System Sci.* 3:1 (1969) 37–56.
3. A. V. AHO AND J. D. ULLMAN, Properties of syntax directed translations, *J. Comput. System Sci.* 3:3 (1969) 319–334.
4. A. V. AHO AND J. D. ULLMAN, Translations on a context-free grammar, *Inform. Contr.* 19:5 (1971), 439–475.
5. N. CHOMSKY, "Syntactic Structures," Mouton, The Hague, 1957.
6. K. CULIK, On some transformations in context-free grammars and languages, *Czechoslovak Math. J.* 17 (1967), 278–311.
7. R. W. FLOYD, On the nonexistence of a phrase structure grammar for ALGOL 60, *Comm. ACM* 5 (1962), 483.
8. E. T. IRONS, A syntax directed compiler for ALGOL 60, *Comm. ACM* 4:1 (1961), 51–55.
9. D. E. KNUTH, Semantics of context-free languages, *Math. Systems Theory* 2:2 (1968), 127–146.
10. P. M. LEWIS AND R. E. STEARNS, Syntax directed transduction, *J. Assoc. Comput. Mach.* 15:3 (1968), 464–488.
11. L. PETRONE, Syntactic mappings of context-free languages, *Proc. IFIP Congress* 2 (1965), 590–591.
12. W. C. ROUNDS, Mappings and grammars on trees, *Math. Surveys Theory* 4:3 (1970), 257–287.
13. A. SALOMAA, "Formal Languages," Academic Press, New York, 1973.
14. J. W. THATCHER, Transformations and translations from the point of view of generalized finite automata theory, *in* "Proc. ACM Symposium on Theory of Computing," pp. 124–142, 1969.
15. J. W. THATCHER, Generalized sequential machines, *J. Comput. System Sci.* 4 (1970), 339–367.
16. V. K. VAISHNAVI AND S. K. BASU, On coupled languages and translations, *Internat. J. Comput. Math. Ser. A* 6 (1977), 33–54.
17. D. H. YOUNGER, Recognition and parsing of context-free languages in time n^3, *Inform. Contr.* 10:2 (1967), 189–208.